signals are not present in the first-level cache, the IP forwards the request address to its local SC across its associated IP/SC interface. The request address and associated function signals ... are latched into a respective input circuit ... in the local SC's input logic circuits.... After the request is latched ... the latched request is provided to the appropriate local segments 0, 1, 2, 3..."

Accordingly, Bauman describes merely forwarding requests to a memory. Applicant's note that the requests are not modified in any manner.

In contrast, Applicant's claimed invention, as amended, now recites "...a memory interface for interfacing with the memory device *wherein one or more of the host applications and the memory device operate in response to different protocols* a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications; and control logic operably coupled *to obtain memory access requests from the number of contexts in a protocol associated with the corresponding host interface, translate the memory access requests into memory access requests in accordance with a protocol of the memory device, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications* ..."

On page 12 of the Office Action, the Examiner responded to Applicant's earlier argument that the host application interface may have different features than the memory interface, stating that the element was not clearly recited in the claims. Applicants have amended the claims to more distinctly highlight the differences in the two protocols, and the translation step that occurs at the interface. Accordingly, Applicant's submit that the claimed invention is patentably distinct over Bauman, (which fails to show both the elements of 'the host applications and memory device operate in response to different protocols' and the step of 'translate the memory access

requests into memory access requests in accordance with the memory device..."), and that therefore the rejection of claim 32 should be withdrawn. Claims 33 – 48 serve to further limit claim 32 and are therefore allowable with claim 32.

Rejections under 35 U.S.C. §103(a)

Claims 1, 3-12, 14, 17, 19-31, 33 and 35-47 were rejected under 35 U.S.C. §103(a) as being unpatentable over Bauman et al (U.S. Pat. 5,875,472) in view of Hughes (U.S. Pat. 5,784,582).

Hughes describes a router having a synchronous dynamic random access memory (SDRAM) based shared memory, with a controller configured to control the order in which the SDRAM access is granted to a plurality of interfaced components. (Abstract).

Hughes describes, at column 2, "... The memory controller stores requests from the plurality of paths, supplies a current request for access to the shared memory pipeline, and selects a next request from the stored request for access to the shared memory pipeline..." No description of suggestion is found in Hughes which overcomes the deficiencies of Bauman. That is, Hughes also neither describes nor suggests the limitations of the claims, such as claim 1, which recites "a host interface for interfacing with the plurality of host applications, control logic operably coupled to obtain memory access requests from the number of contexts in a protocol associated with the corresponding host interface, *translate the memory access requests into memory access requests in accordance with a protocol of the memory device,* interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications, and provide the result/status information to the number of host applications via the number of contexts in accordance with the protocol associated with each of

the number of host applications..." Rather, it appears that all of the interface devices merely forward requests to the controller of Hughes, and they are placed in a pipeline. Accordingly, because the combination of Hughes and Bauman neither describe nor suggest the limitations of claim 1, claim 1 is patentably distinct over the combination, and the rejection should be withdrawn. Dependent claims 2-16 serve to add further patentable limitations to claim 1 and are allowable for at least the reasons put forth with regard to claim 1.

Similar to claim 1, Applicants' claim 17 recites "...host interface logic for interfacing with a plurality of host applications, *the host interface logic operating according to a first protocol ... memory interface logic for interfacing with a memory device, the memory interface logic operating according to a second, different protocol* ... a number of contexts operably coupled to the host interface logic for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications; and *control logic operably coupled to obtain memory access requests from the number of contexts in the first interface protocol, translate the memory access requests into memory access requests in the second interface protocol, interact with the memory device using the memory interface logic for servicing the memory access requests on behalf of the number of host applications,* and provide the result/status information to the number of host applications via the number of contexts in accordance with the first interface protocol ..."

Accordingly, claim 17 is patentably distinct over the combination of Bauman and Hughes, which neither describe nor suggest, in isolation or combination, 'a first protocol...' a 'second protocol...' and 'control logic ... [to] translate ...' For at least this reason, claim 17 is patentably distinct over the combination of references, and the rejection should be withdrawn.

Dependent claims 18-31 serve to add further patentable limitations to parent claim 17, but are allowable for at least the reasons put forth with regard to claim 17.

Claims 2, 16, 18, 34 and 38 were rejected under 35 U.S.C. §103(a) as being unpatentable over Bauman in view of Hughes and further in view of Wentka, U.S. Patent 5,968,114.

Applicant's note that claims 2, 16, 18, 34 and 38 serve to limit parent independent claims which Applicant has indicated are allowable for the reasons stated above.

Applicants have made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Lindsay G. McGuinness, Applicants' Attorney at 978-264-6664 so that such issues may be resolved as expeditiously as possible.

For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

1/26/2004
Date

Lindsay G. McGuinness, Reg. No. 38,549
Attorney/Agent for Applicant(s)
Steubing McGuinness & Manaras LLP
30 Nagog Park Drive
Acton, MA 01720
(978) 264-6664

Docket No. 2494/103 120-034
Dd: 11/26/2003

CLAIMS

B1

1. (Currently amended) A memory interfere device for interfacing a number of host applications to a memory device, the memory interface device comprising:

a host interface for interfacing with the number of host applications;

a memory interface for interfacing with the memory device wherein one or more of the host applications and the memory device operate in response to different protocols ~~have different interface requirements~~;

a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications; and

control logic operably coupled to obtain memory access requests from the number of contexts in a protocol associated with the corresponding host interface, translate the memory access requests into memory access requests in accordance with a protocol of the memory device, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications ~~in accordance with the interface requirements for the memory device~~, and provide the result/status information to the number of host applications via the number of contexts in accordance with the protocol associated with ~~interface requirements for~~ each of the number of host applications.

2. (original) The memory interface device of claim 1, wherein the number of host applications comprises a number of packet processing contexts of a packet processor, and wherein the host interface conforms to a packet processor interface.

3. (original) The memory interface device of claim 1, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface conforms to a CAM interface.

4. (original) The memory interface device of claim 1, wherein the number of contexts comprise a number of context registers sets.

5. (original) The memory interface device of claim 4, wherein each context register set corresponds to one and only one of the number of host applications.

6. (original) The memory interface device of claim 1, wherein the control logic comprises:

monitoring logic;

scheduling logic;

memory interface logic; and

result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device over the memory interface; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

7. (original) The memory interface device of claim 6, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

8. (original) The memory interface device of claim 7, wherein the predetermined register comprises an instruction register.

9. (original) The memory interface device of claim 6, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to pipeline a plurality of memory access requests over the memory interface.

(31) 10. (original) The memory interface device of claim 9, wherein the scheduling logic is operably coupled to determine that a plurality of memory access request conflict and execute at least one of the conflicting memory access requests as an atomic operation.

11. (original) The memory interface device of claim 10, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

12. (original) The memory interface device of claim 6, wherein the result/status logic is operably coupled to correlate result/status information information with its corresponding memory access request.

13. (original) The memory interface device of claim 6, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

14. (original) The memory interface device of claim 13, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

15. (original) The memory interface device of claim 1 embodied as programmed programmable logic device.

16. (original) The memory interface device of claim 1 embodied as an application specific integrated circuit.

17. (Currently amended) Program logic for programming a programmable logic device, the program logic comprising:

host interface logic for interfacing with a number of host applications, the host interface logic operating according to a first interface protocol;

memory interface logic for interfacing with a memory device, the memory interface logic operating according to a second, different protocol ~~wherein one or more of the host applications and the memory device have different interface requirements~~;

a number of contexts operably coupled to the host interface logic for receiving memory access requests from the number of host applications and providing result/status information to the number of host applications; and

control logic operably coupled to obtain memory access requests from the number of contexts in the first interface protocol, translate the memory access requests into memory access requests in the second interface protocol, interact with the memory device using the memory interface logic for servicing the memory access requests on behalf of the number of host applications ~~in accordance with the interface requirements for the memory device~~, and provide the result/status information to the number of host applications via the number of contexts in accordance with the first interface protocol ~~interface requirements for each of the number of host applications~~.

18. (original) The program logic of claim 17, wherein the number of host applications comprises a number of packet processing contexts of a packet processor, and wherein the host interface logic conforms to a packet processor interface.

19. (original) The program logic of claim 17, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface logic conforms to a CAM interface.

20. (original) The program logic of claim 17, wherein the number of contexts comprises a number of context registers sets.

21. (original) The program logic of claim 20, wherein each context register set corresponds to one and only one of the number of host applications.

22. (original) The program logic of claim 17, wherein the control logic comprises:

monitoring logic;

scheduling logic;

memory interface logic; and

result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device using the memory interface logic; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

23. (original) The program logic of claim 22, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

24. (original) The program logic of claim 23, wherein the predetermined register comprises an instruction register.

25. (original) The program logic of claim 22, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to pipeline a plurality of memory access requests over the memory interface.

26. (original) The program logic of claim 25, wherein the scheduling logic is operably coupled to determine that a plurality of memory access requests conflict and execute at least one of the conflicting memory access requests as an atomic operation.

27. (original) The program logic of claim 26, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

28. (original) The program logic of claim 22, wherein the result/status logic is operably coupled to correlate result/status information with its corresponding memory access request.

29. (original) The program logic of claim 22, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

30. (original) The program logic of claim 29, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

31. (original) The program logic of claim 17 embodied in a computer readable medium.

32. (currently amended) An apparatus comprising:

a number of host applications;

a memory device, wherein one or more of the host applications and the memory device have operate using different protocols ~~different interface requirements~~; and

a memory interface device interposed between the host applications and the memory device and operably coupled to receive memory access requests from the number of host applications, translate the memory access requests into requests in accordance with a protocol of the memory device, interact with the memory device on behalf of the number of host applications for servicing the memory access requests ~~in accordance with the interface requirements for the memory device~~, and provide result/status information to the host applications in accordance with the ~~interface requirements for~~ a protocol of each of the number of host applications.

33. (original) The apparatus of claim 32, wherein the memory interface device comprises:

a host interface for interfacing with the number of host applications;

a memory interface for interfacing with the memory device;

a number of contexts operably coupled to the host interface for receiving memory access requests from the number of host applications providing result/status information to the number of host applications; and

control logic operably coupled to obtain memory access requests from the number of contexts, interact with the memory device over the memory interface for servicing the memory access requests on behalf of the number of host applications, and provide the result/status information to the number of host applications via the number of contexts.

34. (original) The apparatus of claim 33, wherein the number of host applications comprises a number of packet processing contexts of a packet processor contexts of a packet processor, and wherein the host interface conforms to a packet processor interface.

35. (original) The apparatus of claim 33, wherein the memory device comprises a content-addressable memory (CAM), and wherein the memory interface conforms to a CAM interface.

36. (original) The apparatus of claim 33, wherein the number of contexts comprises a number of context registers sets.

37. (original) The apparatus of claim 36, wherein each context register set corresponds to one and only one of the number of host applications.

38. (original) The apparatus of claim 33, wherein the control logic comprises:

monitoring logic;

scheduling logic;

memory interface logic; and

result/status logic, wherein:

the monitoring logic is operably coupled to monitor the number of contexts for detecting memory access requests and providing the memory access requests to the scheduling logic;

B1 the scheduling logic is operably coupled to schedule memory access operations for the memory access requests;

the memory interface logic is operably coupled to generate memory interface signals for interfacing with the memory device over the memory interface; and

the result/status logic is operably coupled to provide result/status information to the number of host application(s).

39. (original) The apparatus of claim 38, wherein each context comprises a context register set, and wherein the monitoring logic is operably coupled to monitor a predetermined register in each context register set to detect a memory access request.

40. (original) The apparatus of claim 39, wherein the predetermined register comprises an instruction register.

41. (original) The apparatus of claim 38, wherein the memory interface supports pipelining of memory access operations, and wherein the scheduling logic is operably coupled to a pipeline a plurality of memory access requests over the memory interface.

42. (original) The apparatus of claim 41, wherein the scheduling logic is operably coupled to determine that a plurality of memory access requests conflict and execute at least one of the conflicting memory access requests as an atomic operation.

43. (original) The apparatus of claim 42, wherein the scheduling logic is operably coupled to clear the pipeline in order to execute the conflicting memory access request as an atomic operation.

44. (original) The apparatus of claim 38, wherein the result/status logic is operably coupled to correlate result/status information with its corresponding memory access request.

45. (original) The apparatus of claim 38, wherein the result/status logic is operably coupled to store the result/status information for each memory access request in a corresponding context.

B1

46. (original) The apparatus of claim 45, wherein each context comprises a validity indicator, and wherein the result/status logic is operably coupled to set the validity indicator in each context when the corresponding memory access is complete and the result/status information is available.

47. (original) The apparatus of claim 32, wherein the memory interface device is a programmed programmable logic device.

48. (original) The apparatus of claim 32, wherein the memory interface device is an application specific integrated circuit.